

Attributes for Communication between Scheduling Instances

Status of this Draft

This draft provides information for the grid scheduling community.
Distribution of this memo is unlimited.

Copyright Notice

Copyright (c) The Grid Forum (2001). All rights reserved.

Abstract:

This document describes a set of attributes of a lower level scheduling instance - as found locally on systems - that can be used by a higher level scheduling instance - as found in an Grid environment to interact with remote local scheduling systems. This set of terms provides directions for implementers of new schedulers that are used in computational grids.

1. Introduction

A Computational Grid typically consists of a variety of different resources with different owners. Usually those resources are not exclusively dedicated to Grid usage. For instance, a computer may temporarily be removed from the grid to solely work on a local problem. Many single Grid resources use local management systems that may include separate scheduling instances.

In general, sites can freely participate in Grid computing by offering resources provided that certain conditions like security issues are met. The interaction between those grid resources requires a scheduling layer that uses a different scheduling paradigm than that of local schedulers. While local scheduling usually involves a single scheduling instance that has access to all system information, grid scheduling requires interaction to remote sites and their scheduling instances there. This suggests the use of several scheduling layers for the grid. Although the details of this scheduling architecture has not yet been decided, it is clear that those layers need to exchange information.

In this context, a distinction is made between **lower level scheduling instances** for the local scheduling of resources and **higher level scheduling instances**, that is used for interaction in coordinated scheduling in a

Computational Grid. Both scheduling instances need to work efficiently together in order to make best use of the Grid resources.

Lower level schedulers may be part of queuing systems like PBS, LoadLeveler, LSF, NQS etc. or they may already include scheduling features as those e.g. provided by GARA in Globus.

The purpose of this document is the definition of attributes that describe available features of such a lower level scheduling instance. Those attributes are necessary for the interaction between the different levels of Grid scheduling system as there may be different local schedulers with different features.

Note, that these attributes do not describe the structure and syntax for the resource description, like, for instance, the minimum number of processors or the amount of memory requested. This information can for instance be accessed through the Grid information service. While this is also an important part for the grid infrastructure it is specified in a different document. Instead the features offered by the lower level scheduling instance to the higher level scheduling instance is addressed in this document. This document does not define the interface between the different scheduling layers in detail. However, it can be seen as the first step in this direction.

Note, that the resource features are no obligation or limitation of a lower level scheduling system. For instance, not all kind of features may apply for all resources. Also a new lower level scheduler may offer additional features that are not covered by this list. Therefore this list should serve as a starting point.

3. Typical Scenario

In a typical example, the higher level scheduling instance (a grid job scheduler) that coordinates the scheduling for multi-site application or helps to select the best resources for a job among different possible resource offers. Typically this scheduler itself has no direct control over resources. Therefore, it needs to communicate with and appropriately trigger lower-level scheduling instances. Those lower level schedulers either control resources directly or has some kind of access to their local resources. However, note that the concept is not restricted to two levels of scheduling instances.

The lower level scheduling instance can be local scheduler for a single resource or it may be a scheduling system that manages several resources. The type of resources is not restricted to computing resources but may also include other resources as for example some network bandwidth that is controlled by a bandwidth broker.

In this document we will use the term "allocation" for assignments of resources to a request. The allocation is *tentative* until it is finally executed that is until resources are actually consumed. The schedule maintained by some management instance gives information on the planned or

guaranteed allocations. Also note that an allocation only deals with the guaranteed assignment of resources, but does not guarantee the completion of a job.

As a simple example assume a Computational grid that includes the following resources from different institutions:

- Several clusters of workstations
- a visualization cave
- a special database
- a network with bandwidth brokerage that connects the other resources.

On request of an application a grid job scheduler tries to find a combined allocation that includes a set of 5 workstations, a visualization cave during one stage and the access to a database at another stage. To this end, the grid job scheduler has to interact with the local management instances of the various resources to find suitable allocations. In order to do its job the grid job scheduler needs information about not only the available resources but also the available features of the corresponding local schedulers.

We suggest that this information can be provided in form of a list of attributes.

4. Attributes of allocation properties

These attributes are useful for a higher-level scheduling instance to determine timing and lengths of allocations.

4.1 Allocations run-to-completion

This property states that the local management will not preempt, stop or halt an application after it has been started. The allocation will stay active on the given resources until the end of the requested time or the completion of the job.

4.2 Exclusive Allocations

This property indicates that the allocation runs exclusively on the provided set of resources. The resources are not time-shared and the executed allocation is not affected by the execution and resource consumption of another allocation running concurrently.

4.3 Requirement for providing maximum allocation length in advance (Support/No Support for undetermined allocation length)

This attribute indicates that the local management system requires that an allocation length is given in advance. Historically, resource requests often have been submitted without additional information on the time resources will be used. These programs have been started and run until completion. Current scheduling algorithms as, for example, the backfilling algorithms requires these additional information on maximum allocation length.

4.4 Malleable Allocations

This attribute indicates that the resource-set of an application can change the resource set during execution, that is the local management system supports that resources can be added to or taken from applications during run-time. This modification of the allocation does not require run-time involvement of the higher-level scheduling instance.

Option: Moldable Allocations

This attribute indicates that the local resource management can only increase the resource set of an allocation during run-time. In contrast to malleable allocations resources are not taken from the application.

4.5 Guaranteed completion time of allocations

This feature indicates that the local management system gives guarantees for the completion time of an allocation. This does not necessarily mean that the actual allocation is known but only that the system guarantees that the requested resource allocation has been executed before a given deadline.

4.6 Access to the tentative schedule

This feature indicates that the local management returns on request a current schedule for present and future allocations. This information can be helpful for the grid job scheduler to determine suitable timeslots for instances for co-scheduling in multi-site computing on distributed resources. There are many options possible for this attribute.

Option: Projected starting time of a given allocation

Option: Full/Partial information on the current schedule.

The access to the schedule may be limited due to policies or restrictions of the local management.

5. Attributes for manipulating the allocation execution

These attributes indicate supported functionalities of the lower-level scheduling instance that can be used by a grid job scheduler.

5.1 Preemption

The local resource management allows the temporary preemption of a allocation from the outside. In this case the corresponding application is stopped but remains resident on the resource and can be resumed later. This preemption is not synonymous with the preemption in a multi-tasking system that typically happens in the time range of milliseconds. It only indicates that the local management offers the ability to remotely initiate a preemption of another allocation to e.g. temporary free resources for other usage.

5.2 Migration

The local management system allows the migration of an application or part of an application from one resource set or subset to another set. This way an application can be stopped at one location, the corresponding data moved to another location and the execution can be resumed.

5.3 Remote Co-Scheduling

This attribute indicates that the local management instance allows co-scheduling where the actual resource allocation and schedule is generated by a higher scheduling instance. This includes the generation/cancellation of allocations on the local schedule by the remote management. For instance a grid job scheduler can quickly co-allocate resources at different sites to fulfill a more complex job request.

6. Attributes for requesting resources

6.1 Advanced Reservation

This attribute indicates that advanced reservation is supported according to the proposed advanced reservation protocol.

6.2 Allocation Offers

This indicates that the local resource management supports the generation of resource offers, a potential resource allocation, for a request. For instance if several resources are capable to fulfill a request, a grid job scheduler can first query those system for the allocation and afterwards make its decision to accept the allocation.

Options: Single/Multiple offers for a request.

This option indicates that the local management system can provide several offers for a request with possible overlapping allocations. For instance a grid job scheduler may use this feature for multi-site application where corresponding allocations must be found on different sites.

6.3 Allocation Cost/Objective Information

This attribute indicates that the local management system can return objective or cost information for an allocation.

In case of several allocation offers, a grid job scheduler can for instance use this information for the evaluation.

The cost of a specified allocation usually relates to the policy that is applied by the lower-level scheduling instance. This represents the scheduling objective of the owner of the resource.

7. Authors' Address

Uwe Schwiegelshohn
Uwe.Schwiegelshohn@uni-dortmund.de

Ramin Yahyapour
Ramin.Yahyapour@uni-dortmund.de

Computer Engineering Institute, University Dortmund